AD-A104 142   ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND)   F/G 9/2
FILESTORE MANAGEMENT UNDER THE GEORGE 3 OPERATING SYSTEM.(U)
MAR 81   G D WHITAKER
UNCLASSIFIED   RSRE-MEMO-3353   DRIC-BR-79450   NL

1 OF 1
AD
A104142

END
DATE
FILMED
10 81
DTIC

RSRE

MEMORANDUM No. 3353

ROYAL SIGNALS & RADAR ESTABLISHMENT

FILESTORE MANAGEMENT UNDER THE GEORGE 3 OPERATING SYSTEM

Author: G D Whitaker

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
RSRE MALVERN,
WORCS.

RSRE MEMORANDUM No. 3353

(18) DRIC (19) BR-79450

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 3353

Title: (6) FILESTORE MANAGEMENT UNDER THE GEORGE 3 OPERATING SYSTEM.

Author: (10) G. D. Whitaker

Date: (11) Mar [redacted] 81

SUMMARY

(12) 15

This memorandum describes a new method for controlling the filestore under the George 3 operating system on the ICL 1906S computer at RSRE. The introduction outlines the two major activities to be carried out by a filestore management system. There is a description of the system provided by ICL and a discussion as to why this was inadequate for the particular requirements at RSRE. New ideas are put forward and the implementation of these ideas is described. ~~in chapter 4.~~ The actual algorithm, together with the program for implementing the algorithm and the job for running the new system, is described in detail. ~~in chapter 5.~~ The improvements experienced using the new system are summarised. in chapter 6.

(9) Memorandum rept.,

(14) RSRE-MEMO-3353

This memorandum is for advance information. It is not necessarily to be regarded as a final or official statement by Procurement Executive, Ministry of Defence

409929

RSRE MEMORANDUM 3353

FILESTORE MANAGEMENT UNDER THE GEORGE 3 OPERATING SYSTEM

G D Whitaker

LIST OF CONTENTS

1 INTRODUCTION

Users of the George 3 operating system store their programs, data and other information in files. A type of file called a directory file contains details of all the files owned by a user, including any inferior directory files. Thus information is stored in a hierarchical filestore.

The filestore is held on two media, magnetic disc and magnetic tape. Newly created files are stored on disc and are said to be on-line. Periodically all new or changed files on the discs are copied onto magnetic tape (dump tapes). Eventually the disc space is used up and one of the discs becomes full. At this point a part of the operating system called the backing store unjammer is brought into action to clear some space on the discs. This is achieved by erasing a selection of files from the discs after ensuring that there is a back-up copy on a dump tape. The file is then said to be off-line and before it can be used it has to be retrieved from the tape and a fresh copy made on disc.

Occasionally discs become corrupt and, if any system file has been lost, all the information on the discs has to be reconstituted. When the fault has been rectified, the system will request the latest dump tape containing all the files vital to its operation and will copy all the files on that tape on to the discs. This process is called a general restore. If no vital system file has been lost, for example if the only corruption is to a directory file, it is sometimes only necessary to restore part of the filestore; this is called a partial restore.

A restore will only regenerate a small fraction of the original on-line filestore and, unless more files are brought on-line in an organised fashion, files will be retrieved at random causing long delays. The system does allow for the automatic running of a program - JUGGERNAUT - to organise retrieves after a general restore.

This memorandum describes a new method for controlling both backing store jams and the juggernaut program. These two activities are related by the fact that those files which should be retrieved after a restore are precisely those which should not be thrown off-line during a backing store jam.

2 THE BACKING STORE UNJAMMER

The backing store unjammer relies on, and can be tuned by, the installation parameters BACKJAM, BACKTHRESH, FORMULA and BSINTERVAL. These four parameters are described below.

George considers the on-line filestore to be subdivided into one or more residences. A single residence may not span more than one disc and is of a fixed size. A residence is considered jammed when the percentage of backing store in use on it is greater than BACKJAM. The unjammer is called in by George whenever a residence is jammed. Since the unjamming process involves complete scans of the filestore, and not just of files on the jammed residence, it is economical to try to avoid future jams on other residences by clearing space on them at the same time. Any residence whose percentage of backing store in use is greater than the value of BACKJAM - BACKTHRESH is said to be above threshold, and will be treated by the unjammer.

For each file in the filestore a number can be calculated which is a measure of how desirable it is considered to be to keep that file on-line. This number is referred to as the file's formula. A file whose formula is above the installation parameter FORMULA and which satisfies certain other criteria is a candidate for being thrown off-line by the unjammer. Time is an important factor in calculating a file's formula and George makes use of an inner clock which only runs when the system is running. This clock measures minutes and is referred to as George Mean Time. The value of a file's formula is calculated from

$$\text{formula} = \frac{\text{size}}{64}(\text{GMTSLA} + \text{AVACC})$$

where size is in 512-word blocks,
GMTSLA is the George Mean Time since the file was last accessed
and AVACC is the average George Mean Time between accesses, calculated by

$$\text{New AVACC} = 3/4 \text{ Old AVACC} + 1/4 \text{ GMTSLA}$$

The installation parameter FORMULA is the one installation parameter which is automatically adjusted by the system. A low value of FORMULA means more severe jams (more files have their formula above FORMULA) but a longer time interval between jams (more disc space is cleared in each jam). By changing FORMULA George attempts to make the time between jams equal to BSINTERVAL.

FORMULA is changed in two distinct ways. The first is at the start of a jam when its new value is calculated from

$$\text{FORMULAnew} = \frac{\text{FORMULAold}}{10}\left(9 + \text{minimum of}\left(2, \frac{t}{\text{BSINTERVAL}}\right)\right)$$

where t is the George Mean Time since the last jam. This prevents FORMULA from changing by more than 10% but should cause its value to stabilise as the target BSINTERVAL is being achieved. The second change made to FORMULA is during the unjamming process. After every scan of the filestore which fails to clear the jam the working value of FORMULA for the next scan is reduced by 25%. This value is only valid for the current jam and is forgotten when the jam is cleared. An alternative action after a filestore scan has failed to clear the jam is for a dump to be taken before the next scan, thus ensuring that all files have current magnetic tape copies and can be thrown off-line, but this is extremely rare.

3 CHARACTERISTICS OF THE FILESTORE

From the position in 1971 when all files could be stored on disc, the RSRE filestore increased in size until in 1979 only one fifth would fit on to the available discs (Fig 1). During this period the backing store unjammer was in operation without any serious problems but by this time its job was becoming more critical. As can be seen from Figure 2 using data obtained in April 1979, approximately 100,000 blocks (the size of the on-line filestore) were being accessed every two weeks. Ideally these blocks would be precisely those on-line but the figures showed that a large number were off-line when they were required.

The situation became acute during a sharp rise in the filestore size - 74762 blocks in twelve weeks (6430 blocks/week). These new files were put onto the discs causing backing store jams which threw off-line other files. Many of these files were still being used regularly and so were retrieved, causing more jams. In the summer of 1979 jams were occurring on average 3 or 4 times every day and the number of files being retrieved was approximately the same as the number being thrown off-line. Experience from users of the computer suggested that many of these were the same files but this is difficult to corroborate except for particular instances. The number of jams was disturbing not only because files which were still required were being thrown off-line but also

1) large amounts of processor mill time were being used by the unjammer causing a slow system response time and

2) a large number of magnetic tapes were required causing long delays whilst these were located and loaded.

In the long term the solution would be to increase the disc space but for the short term it was important to make sure that exactly the correct selection of files were on-line to minimise retrieves and in turn reduce the number of jams.

Users of RSRE's computer tend to work in bursts, that is they have a period of intense work on the computer regularly accessing their files and then they work on other non-computing problems. During this time the average access time associated with each of their files is low and so the formula for each file is also low, allowing it to remain on-line. When the user returns to the computer and starts to use the files again the average access time has increased significantly and the files become prime candidates for being thrown off-line. This is precisely opposite to the users expectations and requirements. Therefore more weight should be given to the GMTSLA (time since last access) and less to the AVACC (average access time) when calculating a file's formula.

A study was made to see what effect this different calculation of formula might have and what other improvements could be made. This study took the form of a simulated backing store jam (equivalent to a simulated general restore and juggernaut) with various algorithms being used. The output consisted of a list of files showing which would be on-line and which off-line after the simulated jam and also an overview of the effect on the whole filestore. Interpretation of the results was limited by being unable to follow the effects of an algorithm over a series of jams. The lists of files produced showed the characteristics possessed by files on the discs; this was the main criterion for deciding whether a particular algorithm was better or worse than any other. The overall effect on the filestore had to be watched and the aim was for a linear relationship between the range of formulae and the number of blocks occupied by files having a formula less than or equal to each formula. This line would move up if a lot of files were currently being accessed or created and down if filestore usage was low. A linear relationship would permit the installation parameter FORMULA to be increased or decreased to keep a given number of blocks on-line.

The results indicated that the 'size of file' component used by the unjammer was too severe. In fact, a file which was accessed regularly every day could still be thrown off-line! For example, consider a maximum size file which occupies approximately 500 blocks. There are about 1000 George minutes in an average day and so both the AVACC and GMTSLA (just before it is accessed) of the file are 1000. These figures give 15625 as the file's formula and FORMULA was varying between 5000 and 12000. The ICL manual(1) states that it is "far better to throw off one ten-block file than ten one-block files of similar AVACC and GMTSLA, as it can only lead to one retrieve". This is certainly the case, but a file which is used every day should be kept on-line if at all possible otherwise it will lead to a retrieve every day.

The lists showed that many files had a number of versions also in the filestore. These other versions were a consequence of using the George editor which works by creating a new file and copying text (with some alterations) from the old file to the new one. In most cases the new file is simply a new generation of the old one. Old generations of a file are not often required again, particularly not after an initial period of time. These files are either forgotten or left around in the unlikely event that the new version proves to be worse than the original. This is normally discovered within a short time but even if the old files have been thrown off-line in the meantime a user finds this understandable. The study indicated that a significant improvement in performance could be gained by including this observation in the algorithm and throwing off-line old generations which had not been accessed for a time.

The algorithm which emerged from the study was extremely simple - when a jam occurs, throw off files with the highest GMTSLA. To be practical and avoid filling up the backing store with very large files, it was decided to include some size penalty in the new algorithm.

4 PUTTING IDEAS INTO PRACTICE

If the backing store unjammer were changed and the new version failed to work correctly, filestore information could be lost and the system crash. It is not an easy matter to change George itself and so the strategy adopted was to run the new algorithm as a separate program which could easily be modified or simply not run should anything go wrong. This program would not be called in automatically to clear jams but by running it every night it was hoped to either avoid jams altogether or at least have only one slight jam late in the day when the multi-access terminals had been switched off, and so not affect computer users.

With the introduction of the program, two systems were keeping the discs tidy (the George unjammer was still in operation) and, since different algorithms were used, the new program often found files which had been thrown off-line by George which it would have left on-line. The program had been written to allow the option of retrieving these files (to replace the JUGGERNAUT program) ordered so that files to be retrieved from the same dump tape would be retrieved together. The tapes themselves were ordered so that the tape containing the most files to be retrieved would be requested first. The program would thus attempt to counteract George, but in the event there seemed little point in retrieving files which George would throw off again at the next jam. As the combined system settled down to the new approach there were fewer and less severe jams so that the new program made an increasing contribution to the filestore management.

It was also necessary to tune FORMULA and BSINTERVAL to take account of this new method of working. The night time program run had the effect of a backing store unjam without the actual jam but it did not indicate this to the unjammer by changing FORMULA. This gave rise to two contrasting situations:

1) The program run took place just before a jam was about to occur. When the jam finally did occur it calculated that there had not been a jam for almost twice the usual BSINTERVAL (assuming jams to occur every BSINTERVAL minutes) and so wrongly concluded that it had thrown off-line too many files at the last jam. This caused a rise in FORMULA which resulted in fewer files being thrown off-line (or almost none at all) and eventually, after a few days of this situation, to two almost consecutive jams. The double jam of course reduced FORMULA and all was in order again.

2) The program run took place just after a jam and could find few files to throw off-line. This meant that the unjammer was running almost unaffected and the situation was nearly as bad as not running the program at all. This case was not automatically self-correcting and there were three options to correct it depending on its severity. The first was to ignore it on the grounds that the program might be run before the jam the next day. This was the simplest and most commonly adopted approach. The second option was to raise FORMULA manually, causing fewer files to be thrown off-line at the next jam (but enough to clear it) and allowing the new program to have a larger effect. The third option was to decrease BSINTERVAL. This had a similar effect to raising FORMULA but was used to deal with long term trends.

The overall effect was to reverse the natural usage of the two installation parameters. Thus BSINTERVAL was adjusted until it agreed with the interval between jams and FORMULA was adjusted until it caused just enough files to be found to clear a jam until the program was run.

It would have been possible to arrange for the program to change FORMULA automatically but it was difficult to decide what change should be made and also the two systems quickly settled making changes rarely necessary.

5 THE PROGRAM IN DETAIL

Before describing the program itself it is interesting to describe the sources of information it required.

A program segment for scanning the filestore from the top directories down through their inferiors allowed access to the information about files stored in the directory entries. This was the major source of information and included the George Mean Time that the file was last accessed, the average access time, the size etc.

The most difficult piece of information to acquire was the current George Mean Time. It is stored within George and it would be a security risk to allow a program to have access to this area of store. The way around this problem was to access a file and then see at what time it was accessed. Various files were tried but the best - in terms of security and compactness but not of programming ease - was the 'list of jobs' file held in the directory of the user running the program. This is accessed when the program job is run, and thus its "time of access" gives a sufficiently accurate value of George Mean Time. The only other piece of information required which could not be found from a directory entry was the number of generations of this file above the current one. By buffering directory entries and sorting them before applying the actual algorithm it was possible to count the number of higher generations.

One of the major difficulties experienced by the unjammer is that it does not know what setting of FORMULA will release sufficient blocks. Alternatively, "What setting of FORMULA will achieve the desired filestore size?". By starting with the intention of completing two filestore scans it is possible to calculate a setting on the first pass and use that setting on the second pass. This has the disadvantage of an extra filestore scan but then the original unjammer did not always succeed in freeing the jam after one pass and, even when it did, the resulting on-line filestore size could be so high as to cause another jam within minutes or so low as to under-utilise the discs and cause excessive retrieves. The new algorithm only dealt with a limited number of formulae and so it was possible to pick out files with the maximum formula on the first scan thus making the second scan unnecessary if sufficient space had already been found.

The algorithm used is implemented by two procedures. The first is the procedure which studies all directory entries and makes the decision to ignore the entry, throw it off-line or retrieve it. This procedure calculates the number of blocks associated with each distinct value of formula. These are later made cumulative so that associated with each value of formula is the number of blocks in files having a formula less than or equal to it. The value of the program's FORMULA to obtain any size of on-line filestore within the range covered is then easily found.

The actual formula associated with a file is calculated by the second procedure. This procedure divides files into six types.

Type 1: Special files which must be kept on-line. These include vital system files and the directories themselves.

Type 2: Files which either the computer manager or the user who owns the files have indicated should be put off-line.

Type 3: Files which it is considered highly desirable to have on-line. These include:

a) files recently retrieved but not yet used.
- A file is retrieved so that it may be used, to throw it off-line would only cause another retrieve and annoy the user.

b) the top generation of files occupying less than 5 blocks.
- This is intended to include all files which are George job control macro-commands. Since these files are small their overall effect on the system is not too significant, but keeping macros on-line reduces the double retrieve of a macro being retrieved followed immediately by a retrieve of the program it runs, which would almost certainly be off-line as well.

Type 4: Files accessed in the last 1000 George minutes (slightly longer than the average day).

Type 5: Files with higher generations.

Type 6: All other files.

The formula associated with each file is given by the following Algol outline:

```
IF type 1 THEN 0
ELSF type 2 THEN maximum
ELSF type 3 THEN 1
ELSF type 4 THEN (size in blocks - 1) '/' 100 + 1
ELSF type 5 THEN maximum
ELSE ENTIER (gmtsla * size in blocks/10000) + 1
FI
```

The same job is run to retrieve files after a general restore, to retrieve files after a partial restore or to tidy the filestore. The effect of running this job is controlled by its parameters. One parameter chooses between considering the whole filestore and considering only selected branches after partial restores. Other options include suppressing retrieves or suppressing the throwing of files off-line. When considering the whole filestore, the desired filestore size is supplied through the parameters. The first is the maximum size and is used to select the FORMULA for throwing off-line. The second filestore size parameter is a guide to the minimum size and is used to select the formula for retrieving files. Splitting FORMULA in this way permits greater flexibility and more control. The actual filestore size associated with each of the two values of FORMULA is the greatest size not exceeding that specified by the relevant size parameter. If the filestore is changing in a drastic manner, for instance if many large files are being created, then it is possible that the 'greatest size not exceeding the specified size' is very much less than the specified size. This could cause nearly the whole filestore to be thrown off-line and so a built in safety feature checks that the final filestore size is within 5000 blocks of that specified by the lower desired filestore size parameter. If this condition is not satisifed then the program terminates without throwing off or retrieving any files. The option of continuing (without running the whole job again) is left to the computer manager.

A job is prepared by the program which, when run automatically by the original job, will firstly throw off-line files it has selected and secondly retrieve files. Throwing off before retrieving is essential to avoid a jam being caused by files being retrieved before the space has been made available. The progress of this second job is notified to the operators. When retrieves are present, they are ordered so that the magnetic tape containing the most files to be retrieved is requested first and that containing the least, last. No tape is requested if it contains fewer than five files to be retrieved.

The operations staff are given the option of terminating the retrieves before each magnetic tape is requested. They can also restart the retrieves at any tape or have two or more versions of the retrieve job running simultaneously acting on different tapes. To enable them to decide what to do, two lists of magnetic tapes to be requested together with the number of files to be retrieved from each tape are produced. The first list is in the order the tapes will be requested and the second is in order of increasing tape number.

The basic program currently requires 18K words of store but, if retrieves are also to be permitted, then 78K words are used. This extra store is required to order files and tapes so that retrieves are well organised. It is a compromise between too much store and too many backing store transfers. Retrieves are usually only permitted in exceptional cases like after restores when there are few other jobs running and so 78K is not excessive.

Mill time taken is approximately 45 seconds per filestore scan.

6 CONCLUSIONS

The aims for the new system were to reduce user retrieves and backing store jams. It succeeded in both of these aims. The reduction in the number of user retrieves cannot be measured absolutely as it is not possible to distinguish system issued retrieves for processing old dump tapes from user issued retrieves and since the former frequently outnumber the latter, this information can be misleading. However, evidence from the operations staff does confirm that the new method of working has lead to a dramatic reduction in the number of retrieves. The success of the new system can best be judged by observing the number of backing store jams. It was the backing store unjammer which the new program was primarily intended to aid and it is the jams which use a large amount of central processor time, causing slow response at the multi-access terminals.

As can be seen from Figure 3, the number of jams per day decreased markedly with the introduction of the new system. Some smoothing has been applied to this graph. The point plotted against each day is an average of the number of jams that day with the preceding 9 days. The mean number of jams with the old system was 2.89 and that with the new 0.88. An interesting feature masked by the smoothing is the variance in the number of jams from the mean. With the old system the variance was 4.07 and with the new 0.88. This could be explained by the fact that the unjammer does not know how severe to be. If it is too severe then too many files are thrown off-line and there will be fewer jams until the filestore recovers whilst if it is not severe enough, more jams will occur.

The graph of filestore size over the period (Fig 1) shows that the rate of increase in the summer of 1980 (with the new system) was in fact higher than that causing the problems in 1979, the actual figures being 74762 blocks in

12 weeks (6430 blocks per week) in 1979 compared with 100205 blocks in 14 weeks (7157.5 blocks per week) in 1980.

The new system appears to have met its targets but in addition the same program can be used to retrieve files from the dump tapes in a well ordered manner after either a general or partial restore. This has been used successfully and it is a great advantage to be able to recreate the on-line filestore from the dump tapes.

7 AFTERWORD

At the beginning of this memorandum it was stated that the long-term solution to the backing store problem was to increase the disc space. This was done in June 1980 with new, larger discs. The program was used to fill up the discs to a required size. Unfortunately there were problems with the discs themselves which necessitated reversion to the old discs as well as general and partial restores. Without the new program this would have resulted in turmoil for some weeks but, apart from the disc problems, the filestore had recovered within a day or two.

With the extra backing store the situation is not so acute (until it in turn becomes overloaded) but the program is still run to clear enough space for the following day. So far this has only usually involved one scan of the filestore and the throwing off-line of files with the maximum formula, ie old generations. Even a 500 block file can be allowed to stay on-line, without being used, for over a week before it is thrown off-line!

The average number of jams is currently about one every four weeks but this is expected to rise as the size of the filestore continues to increase.

8 ACKNOWLEDGEMENT

Acknowledgement is due to Miss S G Bond for help in designing the filestore management system and also for suggesting improvements to the text of this memorandum.

REFERENCE

1 GEORGE 3 AND 4 Operation Management 1900 Series, ICL Technical Publication 4438.

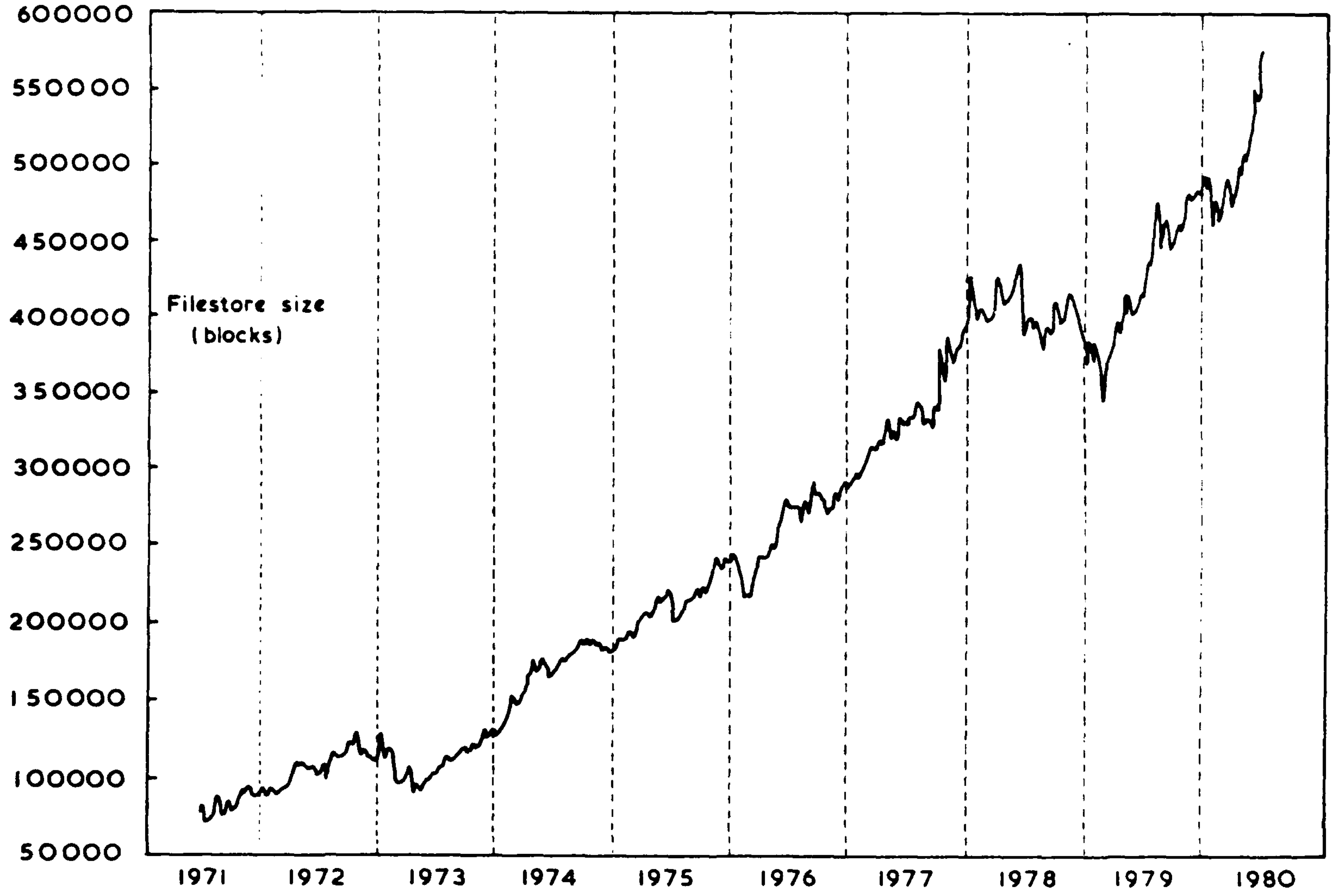

FIG.1. GRAPH OF FILESTORE SIZE 1971-1980

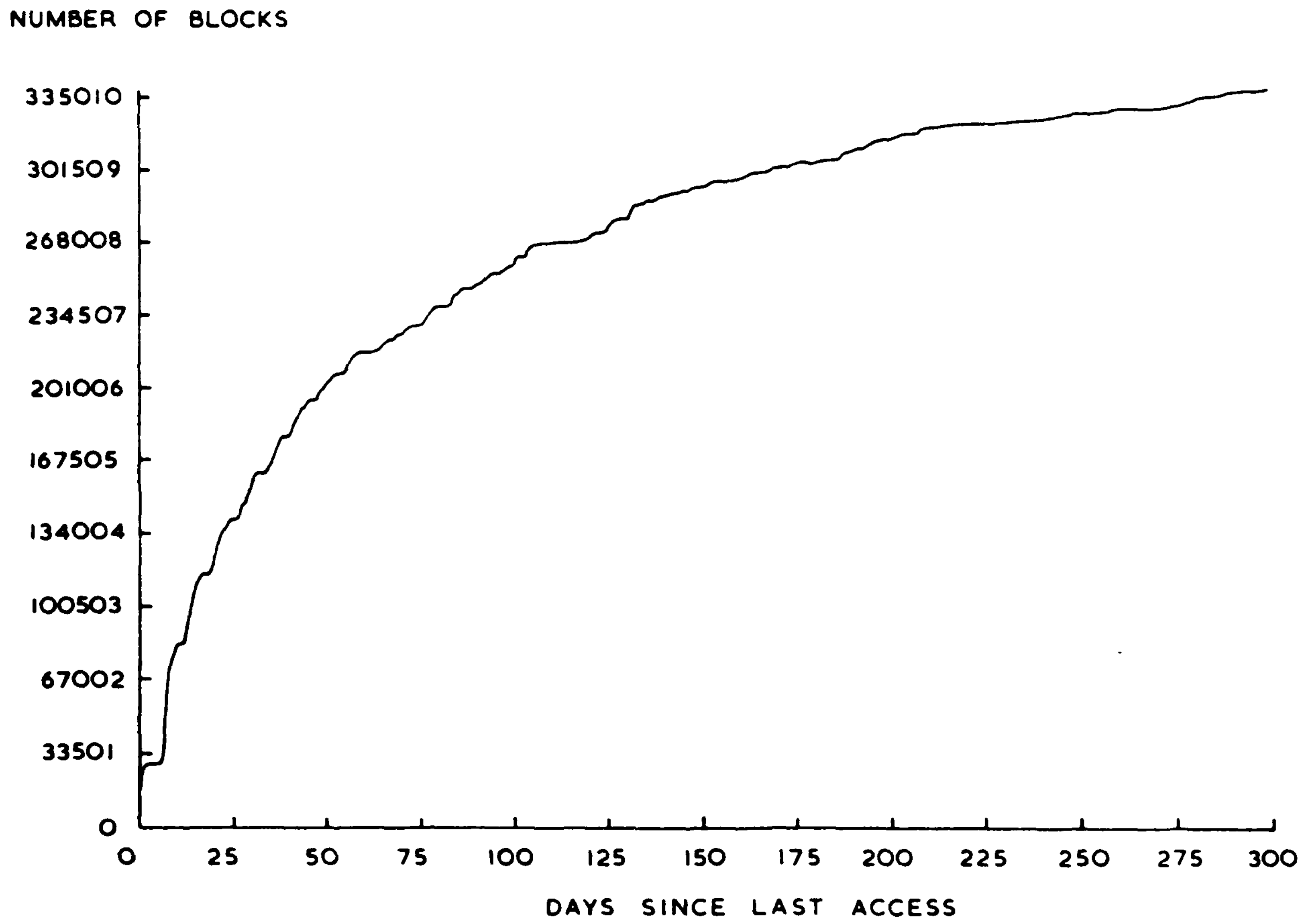

FIG 2 BLOCKS v DAYS SINCE LAST ACCESS

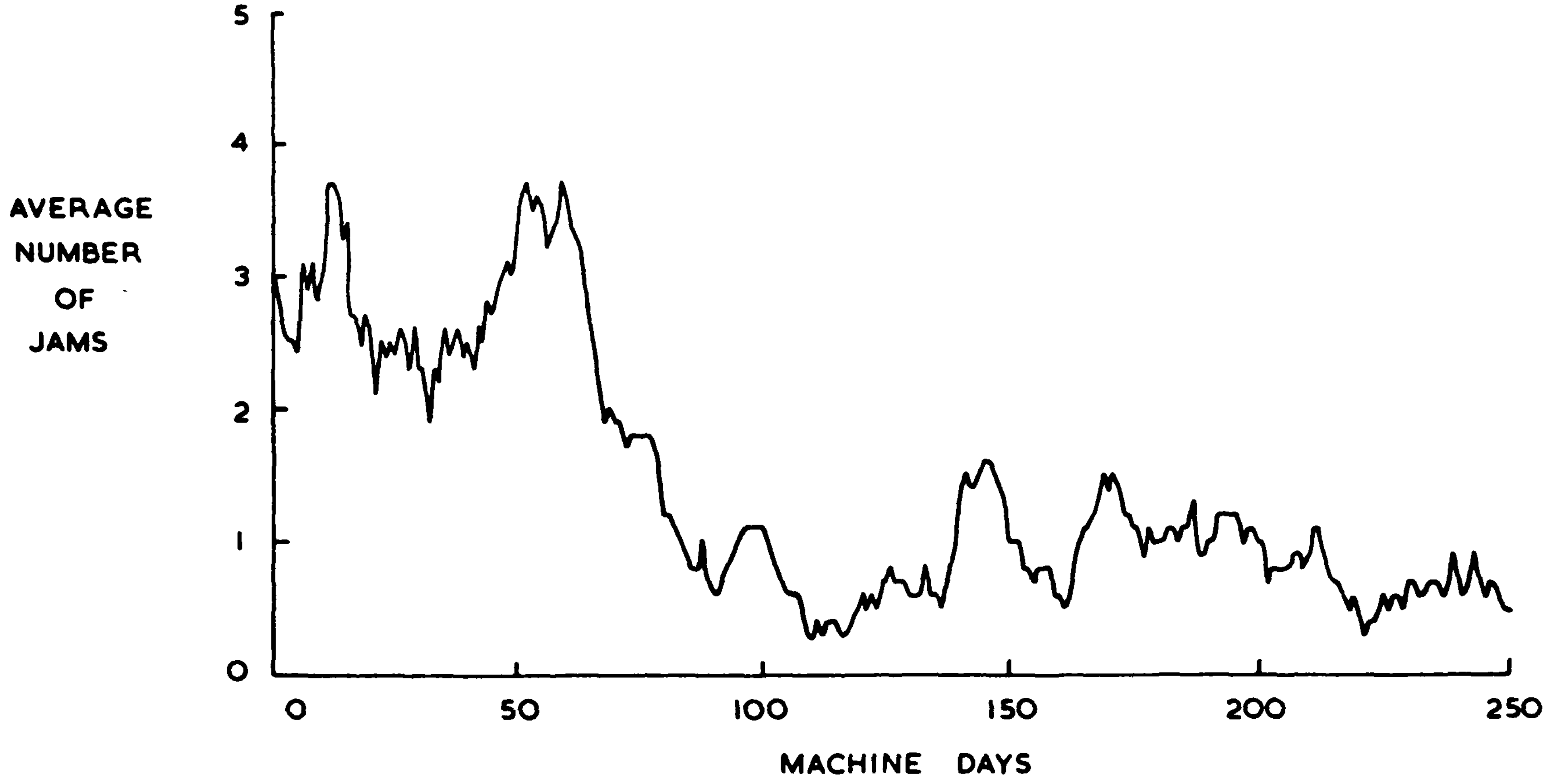

FIG. 3. NUMBER OF BACKING STORE JAMS

14 JUN 79 - 13 JUN 80

DISTRIBUTION LIST

Director	1
Deputy Director	1
Head of Group	1
Superintendent	1
Senior Military Officer	1
Author	5
Library	1
J B Arthur N104 (S)	1
Dr W R Beakley N128 (S)	1
Miss S G Bond N101 (S)	1
Mrs P Cox N114 (S)	1
I F Currie N126 (S)	1
P W Edwards N108 (S)	1
Dr J M Foster N109 (S)	1
R J Granville N113 (S)	1
Ms J M Janicka N118 (S)	1
J D Morison N120(S)	1
N E Peeling N116 (S)	1
Mrs S J Rees N132 (S)	1
Dr C T Sennett N124 (S)	1
ASWE	1
AUWE	1
DOAE	1
RAE	1
RARDE	1
DRIC	3